# Progress Report
# 16-31 December 2011

Amir Hesam Salavati
E-mail: hesam.salavati@epfl.ch

Supervisor: Prof. Amin Shokrollahi
E-mail: amin.shokrollahi@epfl.ch

Algorithmics Laboratory (ALGO)
Ecole Polytechnique Federale de Lausanne (EPFL)

January 11, 2012

## 1 Summary

In the last two weeks, Mr. Amin Karbasi and I were busy extending the approach introduced in [1] to the case of learning a sparse vector which is orthogonal to a set of $m$ patterns of length $n$ with non-negative integer entries, stored in an $\mathcal{S} \times n$ matrix $X$. In what follows, you can find the details of the proposed algorithm and its proof of convergence.

## 2 Leaning a Sparse Null Vector

Suppose we have a matrix $X_{\mathcal{S}\times n}$ composed of non-negative integer entries, where rows of $X$ belong to a subspace with dimension $k < n$. We are interested in finding a vector $w \in \mathcal{R}^n$ such that $X.w = \mathbf{0}$, where $\mathbf{0}$ is the all-zero vector. There are a number of different approaches that can be used to solve this set of equations. However, we are interested in a simple iterative approach that can be implemented by a network of neurons. That's why we focus on the Approximate Message Passing (AMP) algorithm proposed in [1]. There, the authors prove that for a *random Gaussian* matrix $A$, one can recove a sparse vector $u$ from the set of measurements $s = Au$ using AMP and obtain the same optimal sparsity-undersampling trade-off as achieved by using linear programming methods for compressed sensing [1].

Here, we are not interested in the *optimal* solution. A sparse vector $w$ (and not necessarily the sparset such vector) satisfies our needs as we are going to use the given weight vector in the neural associative memory proposed in [2]. Hence, the problem we are interested to solve is

$$\min \|X.w\|_2 \tag{1a}$$

subject to

$$\|w\|_1 \leq q \tag{1b}$$

and

$$\|w\|_2^2 \geq \epsilon \tag{1c}$$

where $q$ determines the degree of sparsity and the constraint (1c) ensures that the algorithm will not converge to the all-zero solution.

### 2.1 The proposed algorithm

To solve the above problem, we first reformulate it as follows:

$$\min \|X.w\|_2 - \lambda(\|w\|_2^2 - \epsilon) \tag{2a}$$

subject to

$$\|w\|_1 \leq q \tag{2b}$$

Now in order to find the solution to problem (2) we propose the following algorithm:

$$y(t) = \frac{X.w(t)}{\|X\|_2} \tag{3a}$$

$$w(t+1) = \eta\left((1+2\lambda_t)w(t) - 2\alpha_t \frac{X^T y(t)}{\|X\|_2}\right)_{\theta_t} \tag{3b}$$

subject to

$$\lambda_{t+1} = \Big[\lambda_t + \gamma(\epsilon - \|w\|_2^2)\Big] \tag{3c}$$

where $t$ denotes the iteration number, $X^T$ is the transpose of matrix $X$, $\gamma$ and $\alpha_t$ are small step sizes, $[.]_+$ denotes $\max(.,0)$, $\theta_t$ is a positive threshold at iteration $t$ and $\eta(.)_{\theta_t}$ is the *point-wise* soft-thresholding function given below:

$$\eta(u)_\theta = \begin{cases} u & \text{if } u > \theta; \\ u & \text{if } u < -\theta \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

In the next section, we derive the necessary conditions on $\alpha_t$ and $\theta_t$ such that algorithm (3) converges to a sparse solution.

## 2.2 Proof of Convergence

Let $E(t) = \|y(t)\|_{\max}$. We would like to show that $E(t+1) < E(t)$ for all iterations $t$. To this end, let us denote $(1+2\lambda_t)w(t) - 2\alpha_t \frac{X^T y}{\|X\|_2}$ by $w'(t)$. Furthermore, let the function $\chi(u;\theta_t)$ be $u - \eta(u)_{\theta_t}$. Rewriting equation (3b) we will have:

$$w(t+1) = w'(t) - \chi(w'(t);\theta_t) \tag{5}$$

Now we have

$$\begin{aligned} E(t+1) &= \|y(t+1)\|_{\max} = \|\frac{X.w(t+1)}{\|X\|_2}\|_{\max} \\ &= \|\frac{X.w'(t)}{\|X\|} - \frac{X.\chi(w'(t);\theta_t)}{\|X\|_2}\|_{\max} \\ &\leq \|\frac{X.w'(t)}{\|X\|_2}\|_{\max} + \|\frac{X.\chi(w'(t);\theta_t)}{\|X\|_2}\|_{\max} \\ &\leq \|\frac{X.w'(t)}{\|X\|_2}\|_{\max} + \frac{\|X\|_{\max}\|\chi(w'(t);\theta_t)\|_{\max}}{\|X\|_2} \\ &\leq \|\frac{X.w'(t)}{\|X\|_2}\|_{\max} + \theta_t\frac{\|X\|_{\max}}{\|X\|_2} \\ &\leq \|\frac{X.w'(t)}{\|X\|_2}\|_{\max} + \theta_t \end{aligned} \tag{6}$$

where the last inequality follows because $\|X\|_{\max} \leq \|X\|_2$. Now expanding $\frac{X.w'(t)}{\|X\|}$ we will get

$$\frac{X.w'(t)}{\|X\|_2} = (1+2\lambda_t)y(t) - 2\alpha_t\frac{XX^Ty}{\|X\|_2^2} = \left[(1+2\lambda_t)I_{\mathcal{S}\times\mathcal{S}} - 2\alpha_t\frac{XX^T}{\|X\|_2^2}\right]y(t) \tag{7}$$

Denoting the matrix $(1+2\lambda_t)I_{\mathcal{S}\times\mathcal{S}} - 2\alpha_t\frac{XX^T}{\|X\|_2^2}$ by $C_t$, we can further simplify inequality (6):

$$\begin{aligned} E(t+1) &\leq \|C_t y(t)\|_{\max} + \theta_t \\ &\leq \|C_t\|_{\max}\|y(t)\|_{\max} + \theta_t \\ &= \|C_t\|_{\max}E(t) + \theta_t \end{aligned} \tag{8}$$

Therefore, in order to show that $E(t+1) < E(t)$, we must have $\|C_t\|_{\max} < 1$ and $\theta_t \to 0$ as $t \to \infty$. By setting $\theta_t = 1/t$ we will achieve the second requirement. Now to ensure $\|C_t\|_{\max} < 1$, we would like to have $|c_{ij}^t| < 1$ for all elements $(i,j)$ of $C_t$. Therefore, by letting $A = XX^T$ we must have the following relationship for diagonal elements:

$$|1 + 2\lambda_t - 2\alpha_t A_{ii}/\|A\|_2| < 1 \Rightarrow \lambda_t < \alpha_t A_{ii}/\|A\|_2 < 1 + \lambda_t,\ \forall i = 1, \dots \mathcal{S} \tag{9}$$

Since $\lambda_t \geq 0$ for all $t$ and $0 < A_{ii} \leq \|A\|_2$, the right hand side of the above inequality is satisfied if $\alpha_t < 1$. The left-hand side is satisfied for $\alpha_t > \lambda_t/a_{\min}$, where $a_{\min} = \min_i A_{ii}/\|A\|_2$. If $\lambda_t = 0$, this is simply equivalent to having $0 < \alpha < 1$.

## 2.3 Simulation Results

We have simulated the proposed approach over three different network sizes $n = 90, 300, 600$ with $m = 60, 200, 400$ constraints, respectively. To execute the learning phase, we first generate $\mathcal{S}$ patterns that belong to a subpace with dimension $k = n - m$ by multiplyig $\mathcal{S}$ *binary* vectors of length $k$ to a *sparse* randomly generated 0/1 matrix $G \in \mathcal{R}^{k \times n}$ (row degree = 6 amd column degree = 2). Then we store all of the generated pattern in the matrix $X_{\mathcal{S} \times n}$. We repeat the learning algorithm given by equations (3) until $\|y(t)\| \leq \delta$, where $\delta$ is a very small number. Furthermore, $\lambda_t$ was bounded above so that it was always less than $a_{\min}/(a_{\max} - a_{\min})$ to ensure that one $\alpha_t$ works for all diagonal element in equation (9).

Table 2.3 summarizes other simulation parameters.

Table 1: Simulation parameters

| Parameter | Maximum Learn Iterations | Number of Learned Patterns $\mathcal{S}$ | $\delta$ | $\theta_t$ | $\alpha_t$ | $\epsilon$ |
|---|---|---|---|---|---|---|
| Value | 100000 | 10000 | $10^{-3}$ | $0.35/t$ | $\begin{cases} 0.9 & \text{if } \lambda_t = 0; \\ \min(\frac{\lambda_t}{a_{\min}}, 1 + \lambda_t) & \text{if } \lambda_t > 0. \end{cases}$ | 0.01 |

We generated 100 different random ensembles, i.e. 100 different random generator matrix $G$, and observed that the proposed algorithm was able to converge to a relatively sparse solution in all 100 occassions. Figure 1 shows the number of iterations executed before convergence is reached for different constraint nodes. We have investigated three different network sizes, i.e. $n = 90, 300, 600$. The corresponding number of constraints were $m = 60, 200, 400$, respectively.

Figure 2 illustrates the percentage of trials with the specified sparsity measure defined as $\rho = \kappa/n$, where $\kappa$ is the number of non-zero elements. From the figure one notices that as $n$ increases, the weight vectors become sparser.

Finally, figure 3 shows the MSE, given by the $\|y(t)\|_2$ in equation (3a), in each iteration for a sample trial for three different values of $n$. Overall, the MSE is decreasing. However, small increases happens due to the addition of the sparsity constraint. The amplitude of fluctuations (and in fact the convergence of the algorithm) depends heavily on the choice of parameters $\alpha$ and $\lambda$.

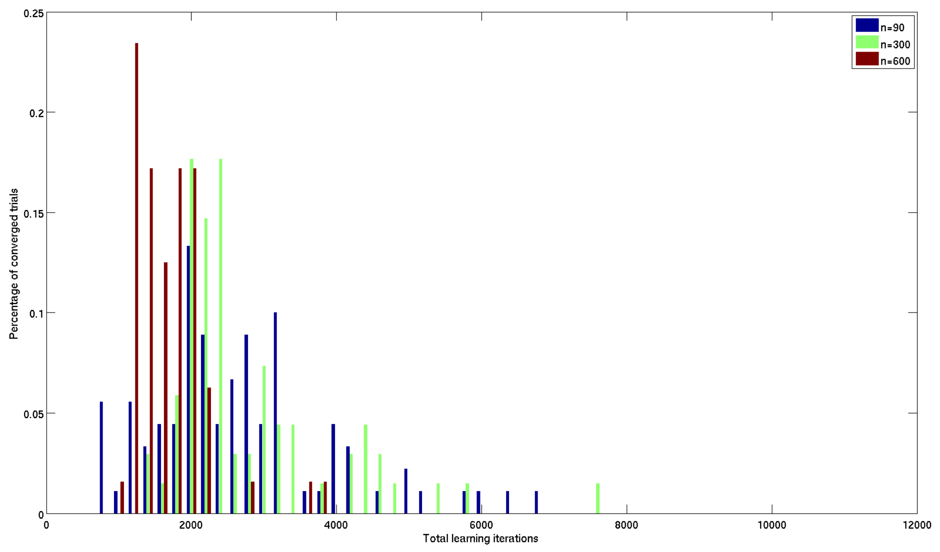

Figure 1: Convergence rate for the 100 trials and different values of network sizes: $n = 90, 300, 600$.

## 3 Future Works

There still remains some technicalities about the proof we discussed in this report. For one, if $\lambda_t$ is not bounded, there can be no $\alpha_t$ that works for all diagonal elements in equation (9). So we have to find a property of the weight matrix $X$ such that this phenomenon does not happen.

Another way to overcome the above problem is to modify the algorithm by considering an additional normalizing term similar to one proposed in [3] which makes the update in each iteration orthogonal to the current vector. As a result, $\|w(t+1)\|_2 \geq \|w(t)\|_2$ in each iteration $t$. Hence, if $\|w(0)\|_2 \neq 0$, we are sure that the final weight vector will not be equal to zero. This is a subject of our future research.

## References

[1] D. L. Donoho, A. Maleki, A. Montanari, *Message passing algorithms for compressed sensing*, Proc. Nat. Acad. Sci., Vol. 106, 2009, pp. 1891418919.

[2] K.R. Kumar, A.H. Salavati and A. Shokrollahi, *Exponential pattern retrieval capacity with non-binary associative memory*, Proc. IEEE Information Theory Workshop, 2011.

[3] L. Xu, A. Krzyzak, E. Oja, Neural nets for dual subspace pattern recognition method, Int. J. Neur. Syst., Vol. 2, No. 3, 1991, pp. 169-184.

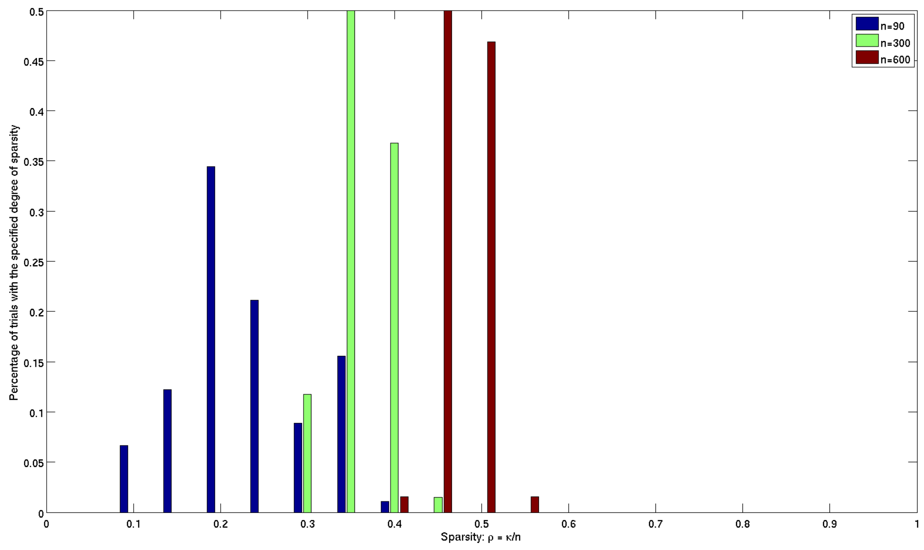

Figure 2: The percentage of trials with the specified sparsity measure and different values of network sizes: $n = 90, 300, 600$. The sparsity measure is defined as $\rho = \kappa/n$, where $\kappa$ is the number of non-zero elements.

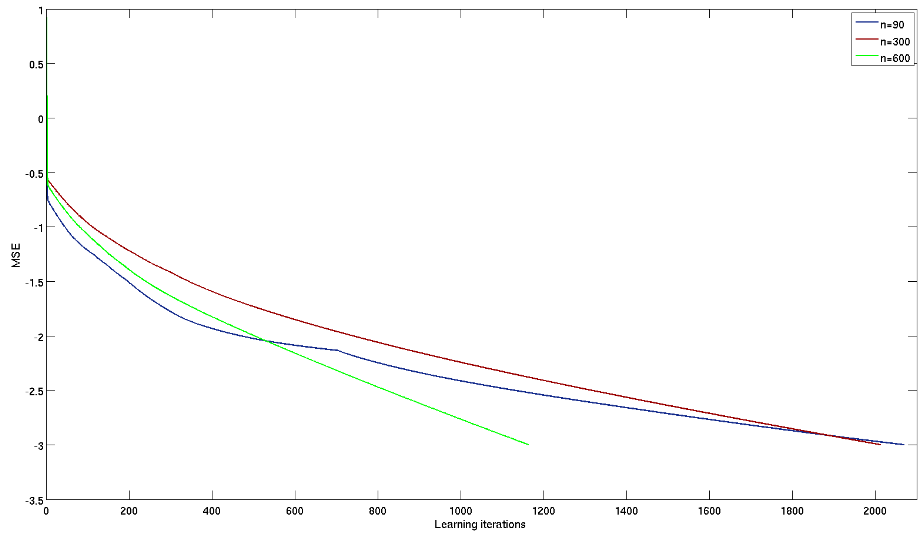

Figure 3: MSE as a function of iteration for a random trial node and three different values of $n$.